

MySQL Query Browser

MySQL Query Browser

Copyright © 2004 MySQL AB

This manual is NOT distributed under a GPL style license. Use of the manual is subject to the following terms:

- Conversion to other formats is allowed, but the actual content may not be altered or edited in any way.
- You may create a printed copy for your own personal use.
- For all other uses, such as selling printed copies or using (parts of) the manual in another publication, prior written agreement from MySQL AB is required.

Please email docs@mysql.com for more information or if you are interested in doing a translation.

Table of Contents

1. Introduction to the MySQL Query Browser	1
2. Installing MySQL Query Browser	2
2.1. Introduction	2
2.2. Installing Under Windows	2
2.3. Installing Under Linux	2
3. Starting MySQL Query Browser	4
3.1. Introduction	4
3.2. Connection Dialog	4
4. A Tour of the MySQL Query Browser	7
4.1. The Main Query Window	7
4.2. The Query Toolbar	8
4.2.1. The Navigation Buttons	8
4.2.2. The Query Area	8
4.2.3. The Action Buttons	9
4.3. The Advanced Toolbar	9
4.4. The Result Area	10
4.5. The Object Browser	10
4.5.1. The Database Browser	10
4.5.2. The Bookmark Browser	12
4.5.3. The History Browser	12
4.6. The Information Browser	12
4.6.1. The Parameter Browser	12
4.6.2. The Syntax Browser	12
4.6.3. The Function browser	12
4.6.4. The Transaction Browser	13
4.7. The Script Editor	13
4.7.1. The Script Debugging Buttons	14
4.7.2. The Script Area	14
5. Using The MySQL Query Browser	15
5.1. Entering and Editing Queries Manually	15
5.2. Building Queries	15
5.3. Creating Views	16
5.4. Navigating Result Sets Within the Result Area	17
5.5. Exporting Result Sets	17
5.6. Handling BLOB and TEXT Columns	17
5.7. Editing Result Sets Within the Result Area	18
5.8. Comparing Result Sets	18
5.9. Creating Master-Detail Views	19
5.10. Editing Queries From a Development Tool	20
6. The MySQL Table Editor	21
6.1. Introduction	21
6.2. The Main Editor Window	21
6.3. The Columns And Indices Tab	22
6.3.1. The Column Editor	22
6.3.2. The Detail Area	22
6.4. The Table Options Tab	24
6.5. The Advanced Options Tab	24
6.5.1. The Various Section	24
6.5.2. The Row Options Section	24
6.5.3. The Storage Options Section	24
6.5.4. The Merge Table Options Section	25
6.5.5. The Table RAID Settings Section	25
6.6. Applying Your Changes	25

7. Options Dialog	27
7.1. Introduction	27
7.2. General Options	28
7.3. Connections	28
7.4. Editors	30
7.5. The Browser Section	31
7.5.1. Display Options	31
7.5.2. Query Options	31
7.5.3. Various	31
A. Installing From Source	32
A.1. Introduction	32
A.2. Downloading The Source Code	32
A.3. Building from Source under Windows	32
A.3.1. Prerequisites	32
A.3.2. Compiling and Building	33
A.4. Building from Source under Linux	35
A.4.1. Prerequisites	35
A.4.2. Compiling and Building	35
B. Troubleshooting Application Errors	38
B.1. Troubleshooting Connection Errors	38
B.2. Troubleshooting Display Problems	38
C. How Connections Are Stored	40
D. XML Files Common to the MySQL GUI Applications	41
E. XML Files Used by MySQL Query Browser	42
F. Notes for Translators	43
G. Third-party Software Used by the MySQL GUI Tools	44
G.1. PCRE Library	44
G.2. PNG Support	44

List of Figures

3.1. Connection dialog	4
4.1. The MySQL Query Browser main query window	7
4.2. The query area expanded to ten lines	8
4.3. The MySQL Query Browser status icons	9
4.4. The advanced toolbar	9
4.5. The database browser	10
4.6. The script editor	13
5.1. The table tool	16
5.2. The BLOB handling icons	18
5.3. The dynamic parameter list	19
6.1. The MySQL Table Editor	21
6.2. The column editor	22
6.3. The index editor	23
6.4. The Confirm Table Edit dialog	25
7.1. Options dialog	27
7.2. Options dialog: Connections	28
B.1. Error message dialog	38

Chapter 1. Introduction to the MySQL Query Browser

The MySQL Query Browser is a graphical tool provided by MySQL AB for creating, executing, and optimizing queries in a graphical environment. Where the MySQL Administrator [<http://www.mysql.com/products/administrator/>] is designed to administer a MySQL server, the MySQL Query Browser is designed to help you query and analyze data stored within your MySQL database.

While all queries executed in the MySQL Query Browser could also be performed in the **mysql** command-line utility, the MySQL Query Browser allows for the querying and editing of data in a more intuitive, graphical manner.

MySQL Query Browser is designed to work with MySQL versions 4.0 and higher.

MySQL Query Browser is to a large extent the result of feedback MySQL AB has received from many users over a period of several years. However, if you find it's lacking some feature important to you, or if you discover a bug, please use our MySQL Bug System [<http://bugs.mysql.com>] to request features or report problems.

Chapter 2. Installing MySQL Query Browser

2.1. Introduction

The MySQL Query Browser is available for both Windows and Linux, in source and binary forms. MySQL Query Browser can be downloaded from The MySQL web site. [<http://dev.mysql.com/downloads/query-browser/>]

2.2. Installing Under Windows

MySQL Query Browser runs on recent 32-bit Windows NT based operating systems, including Windows 2000, XP, and 2003.

MySQL Query Browser is installed through the use of a Windows Installer (.msi) installation package, which can be used on all Windows operating systems. The MSI package is contained within an archive named `mysql-query-browser-version-win.zip`, where *version* indicates the MySQL Query Browser version. Download the ZIP file, extract it and change into the directory you extracted it into.

The Windows Installer system was updated with the release of Windows XP; those using an older version of Windows can reference this Microsoft Knowledge Base article [<http://support.microsoft.com/default.aspx?scid=kb;EN-US;292539>] for information on upgrading to the latest version.

To install MySQL Query Browser, right-click on the MSI file and select Install. The installation will begin automatically after the installer prompts you for your installation preferences. During installation, you can choose whether you want the installer to place a shortcut in the Start menu and an icon on the desktop.

If you are having problems running the installer, you can download a ZIP file without an installer as an alternative. That file is called `mysql-query-browser-version-win-noinstall.zip`. Using a ZIP program, unpack it to a directory of your choice. You may want to create shortcuts to `MySQLQueryBrowser.exe` for your desktop or quick launch bar.

Unless you choose otherwise, MySQL Query Browser is installed in `C:\%PROGRAMFILES%\MySQL\GUITools\MySQL Query Browser\MySQLQueryBrowser.exe`, where `%PROGRAMFILES%` is the default directory for programs on your machine. For example, this directory might be `C:\Program Files` or `C:\programme`.

2.3. Installing Under Linux

MySQL Query Browser runs on Linux machines that have a graphical desktop installed. It is designed to run under the Gnome desktop, but you may also use other desktops, such as KDE, as long as GTK2 is installed. It has been tested on Linux kernel versions 2.4 and 2.6, but it should also run on other versions, and even on a number of Unix-like operating systems.

To install MySQL Query Browser, first download the installation tarball. The tarball file is called `mysql-query-browser-version-linux.tar.gz`, where *version* indicates the MySQL Query Browser version (for example 1.0.5-alpha).

To see all files in the tarball, run this command:

```
shell> tar -tzf mysql-query-browser-version-linux.tar.gz
```

To install MySQL Query Browser, run this command:

```
shell> tar --directory=/opt -xzf mysql-query-browser-version-linux.tar.gz
```

This installs the application binary in `/opt/mysql-query-browser/bin`. Change into that directory and run `mysql-query-browser` to start the application. You can replace `/opt` with your desired installation path.

Distribution-specific packages will be available at some point.

Chapter 3. Starting MySQL Query Browser

3.1. Introduction

The way you start MySQL Query Browser depends on the operating system you are using:

- On Windows, start MySQL Query Browser by double-clicking its desktop icon, or by selecting its entry from the Start menu (typically the MySQL Query Browser entry of the MySQL section of the Start menu). Alternatively, you can open a DOS window and start the MySQL Query Browser from the command line:

```
C:\%PROGRAMFILES%\MySQL\GUITools\MySQLQueryBrowser.exe
```

`%PROGRAMFILES%` is the default directory for programs on your machine. For example, `C:\program files` or `C:\programme`. If your path contains spaces, you should enclose the command within double quotes. For example:

```
C:\> "C:\program files\MySQL\GUITools\MySQLQueryBrowser.exe"
```

- On Linux desktops, start MySQL Query Browser by changing directories into `/opt/mysql-query-browser/bin`, and then executing `mysql-query-browser`.

3.2. Connection Dialog

Once MySQL Query Browser has been started, it displays a connection dialog. You have to specify the MySQL server to which you would like to connect, the credentials needed for authorization on that server, which machine that server runs on (and which port it listens to), and the default database (Schema) you will be querying from. You may also specify a number of other options, if required.

You must choose a default database in order to issue queries. It is possible to choose a default database after connecting to the server, but setting the default from the connection dialog can save time on subsequent connections.

Figure 3.1. Connection dialog

MySQL Query Browser

Connect to MySQL Server Instance

Connection: VBMYSQL

Username: root

Password:

Hostname: localhost

Port: 3307

Schema:

Advanced Connection Options

☒ Use compression protocol

☐ Return number of found rows, not number of affected rows

☐ Ignore spaces after function names, make them reserved words

☐ Enable LOAD DATA LOCAL handling

Number of seconds of inactivity allowed before disconnect

<< Details OK Clear Cancel

If the server connection is successfully established, all of the values filled in the fields of the connection dialog is saved for future connections (see the section that describes how MySQL Query Browser stores connection information). The Password field, however, is always empty: For security reasons, the password is not stored along with the other options, unless you explicitly specify otherwise in the General Options section of the Options dialog.

If you have difficulties connecting, please see the Troubleshooting Connection Errors section in the Troubleshooting Application Errors appendix.

You can change any of the values in the connection profiles just by overwriting the existing values with new ones. Similarly, if you select another connection profile, you can change any of its values. When you click the OK button after changing a profile's values, the changes are stored permanently if a suc-

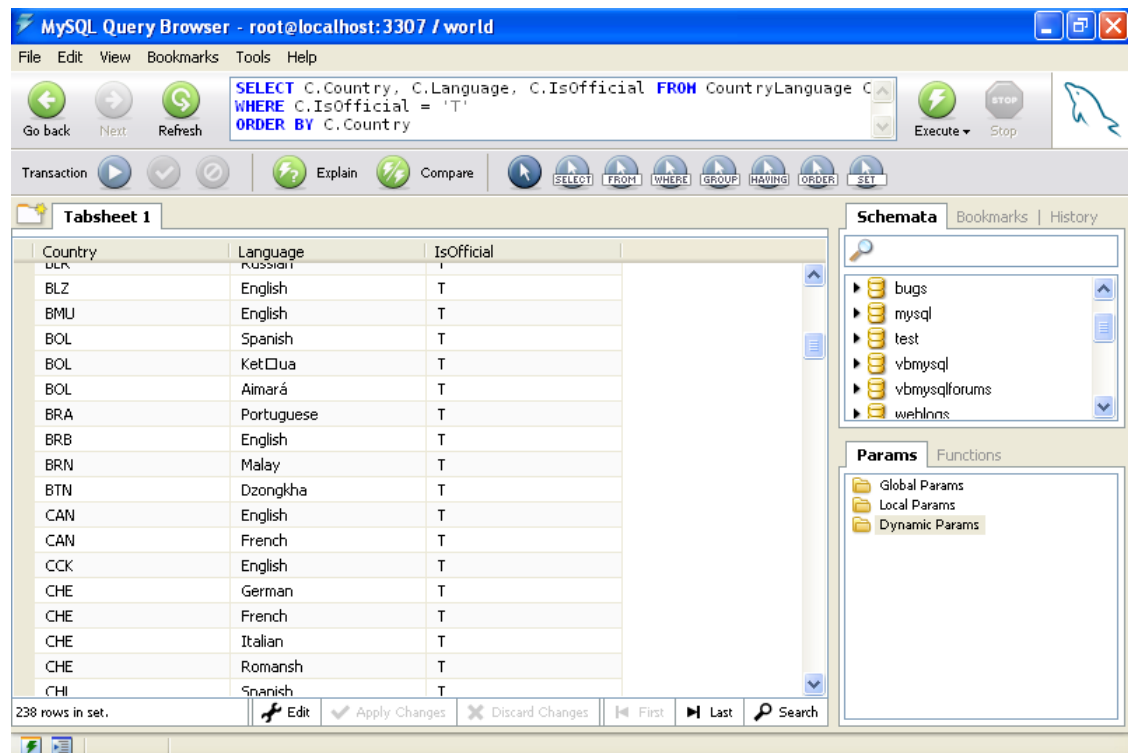
successful connection to the MySQL server is established. You can also click the ... button next to the Connection drop-down box. This brings up an Options dialog window that has a Connections section for modifying connection profiles. You can find the values you can set in the Connection dialog window in the description of the Connections section of the Options dialog.

Chapter 4. A Tour of the MySQL Query Browser

4.1. The Main Query Window

Once you successfully connect to a MySQL server you are presented with the main query window of the MySQL Query Browser. All of the application's functionality is available through this window.

Figure 4.1. The MySQL Query Browser main query window



The main query window is divided up into several sections:

- *Query Toolbar:* The query toolbar is where you create and execute your queries. It is composed of three navigation buttons (Go Back, Next, Refresh), the query area, two action buttons (Execute and Stop), and a status indicator.
- *Advanced Toolbar:* The advanced toolbar contains three sets of buttons: the Transaction Buttons (Start, Commit, Rollback), the query management buttons (Explain, Compare), and the query building buttons (Select, From, Where, and so on.)
- *Result Area:* All query results are displayed in the result area. You can have multiple tabs active at one time, allowing you to work on multiple queries. The result area can be split vertically and horizontally for performing comparisons, and queries in different parts of a split result area can be joined together for master-detail analysis.

- *Object Browser:* The object browser is part of the sidebar and allows you to manage your databases, bookmarks, and history. You can choose which database and tables to query, add commonly-used queries to a collection of bookmarks, and browse through previously issued queries to use them again.
- *Information Browser:* The Information Browser is part of the sidebar and is used to look up built-in functions, and to manage queries that contain parameters. One example of this is when establishing a master-detail pair of queries. The data in the master query is available as parameters to the detail query.

Most of these sections can be displayed and hidden using the View menu by checking and unchecking the elements you wish to view.

Each of these topics are covered in more detail in the sections that follow.

4.2. The Query Toolbar

All queries, whether generated automatically, graphically, or manually, appear in the Query Toolbar. The simplest way to use the MySQL Query Browser is to type a query into the query area and click the execute button. A statement terminator such as `;` or `\G` is not required.

4.2.1. The Navigation Buttons

To the left of the query area are the navigation buttons. The navigation buttons allow you to browse through your query history so that you can review and re-execute a previously executed query. The query history is available through the object browser.

Clicking the Go Back button loads the previous query in your history, whereas the Go Next button loads the following query. Only queries that execute without errors are added to your history.

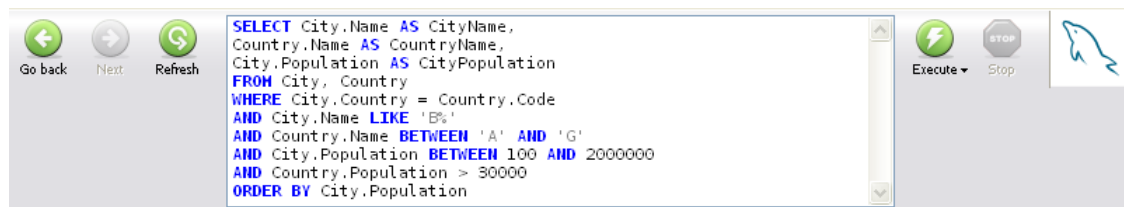
As you navigate with the Next and Go Back buttons, the queries you navigate through are not executed unless you explicitly execute them by clicking the Execute button. The Refresh button re-executes the last executed query, which may or may not be the current query in the query area.

4.2.2. The Query Area

The query area is where the text of all queries and statements are displayed. The query area is three lines high by default and automatically expands to a maximum of ten lines in height. For queries longer than ten lines, the query area is scrollable.

For additional space, you can press the **F11** key to maximize the query area. You can also choose the Maximize Query Area of the View menu to maximize the query area. When the query is maximized, line numbers are displayed for the query, and the query area can be resized by clicking and dragging the line that divides the query area from the result area. To restore the query area, press the **F11** key again.

Figure 4.2. The query area expanded to ten lines



The following commands are available by right-clicking the query area: Cut (**Ctrl+X**), Copy (**Ctrl+C**), Paste (**Ctrl+V**), Clear (**Ctrl+N**), Open Query ... (**Ctrl+O**), and Save Query As ... (**Ctrl+S**). The Open and Save commands load the contents of an SQL file into the query area or save the contents of the query area into a text file.

4.2.3. The Action Buttons

To the right of the query area are the action buttons. The Execute button executes any queries in the query area, whereas the Stop button ceases execution.

If you click the down-arrow below the Execute button, there are three potential execution options:

- **Execute (Ctrl+Enter)**: This executes the query and displays the results in the currently active result area.
- **Execute in new Tab (Ctrl+Shift+Enter)**: This executes the query and displays the results in a newly created result area.
- **Split Tab and Execute (Ctrl+Alt+Enter)**: This splits the result area horizontally and then displays the query results in the lower half of the result area.

To the right of the action buttons is the status indicator. The status indicator shows the standard MySQL logo when the MySQL Query Browser is idle, and displays an animated logo when the MySQL Query Browser is busy processing a query. Both icons can be seen in the following illustration:

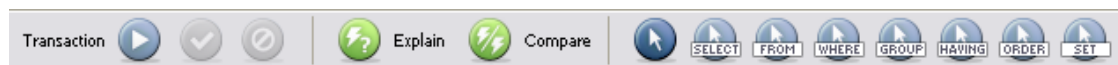
Figure 4.3. The MySQL Query Browser status icons



4.3. The Advanced Toolbar

Below the query bar is the Advanced Toolbar. The Advanced Toolbar contains a set of buttons for transaction control, query management, and query building.

Figure 4.4. The advanced toolbar



The left panel of the advanced toolbar contains the transaction control buttons. From left to right, the buttons allow you to start, commit, and roll back a transaction. As when using the command-line client, you can only use transactions with table handlers that support them (InnoDB for example). More information on transactions can be found in the the MySQL Reference Manual [http://dev.mysql.com/doc/mysql/en/InnoDB_transaction_model.html].

The center panel provides buttons for query management. The Explain button can be used to get the EXPLAIN output for the current query from the MySQL server, whereas the Compare button allows you

to compare the results of two queries.

The next panel contains the query building buttons. You can use these buttons to build a query visually by clicking on the tables and columns that you wish to involve in your query, using specialized mouse pointers to indicate which part of the query the different fields and tables occupy.

The right panel contains the Create View button. More information on the Create View button can be found in the Creating Views section.

4.4. The Result Area

All query results are displayed in the result area. Within the result area you can use multiple tabs, and individual result areas can be split either vertically or horizontally. New tabs can be created either by choosing the Execute in new Tab option on the query bar, or by clicking the New Tab button at the top of the result area. In addition, you can also right-click on the current result area and choose the Add new Tabsheet (Ctrl+T) option.

Individual tabs can be closed by either right-clicking within the result area and choosing the Remove Tab option, or by clicking the red X icon on the tab you wish to close.

To split a result area, right-click on it and choose either the Split Tab vertically or the Split Tab horizontally option. After splitting the result area you can then choose one half of the result area and view result sets within it. You can remove sections of the result area by right-clicking on the section you wish to remove and clicking Remove Resultset.

When navigating a particularly large result set, you can press the **F12** key to maximize the result area. You can also choose the Maximize Tabsheets option of the View menu to maximize the result area. Pressing **F12** again will restore the result area to its original size.

The result area can be used to review and edit the results of a query, with editing permitted as long as the query is based on a single table and there is sufficient key information to uniquely identify rows. To edit the contents of the result area you must enable edit mode through the use of the Edit button at the bottom of the result area. Any edits you make are not immediately applied, but instead you need to click the Apply Changes button next to the Edit button. Clicking the Discard Changes button throws away any changes you have made to the data.

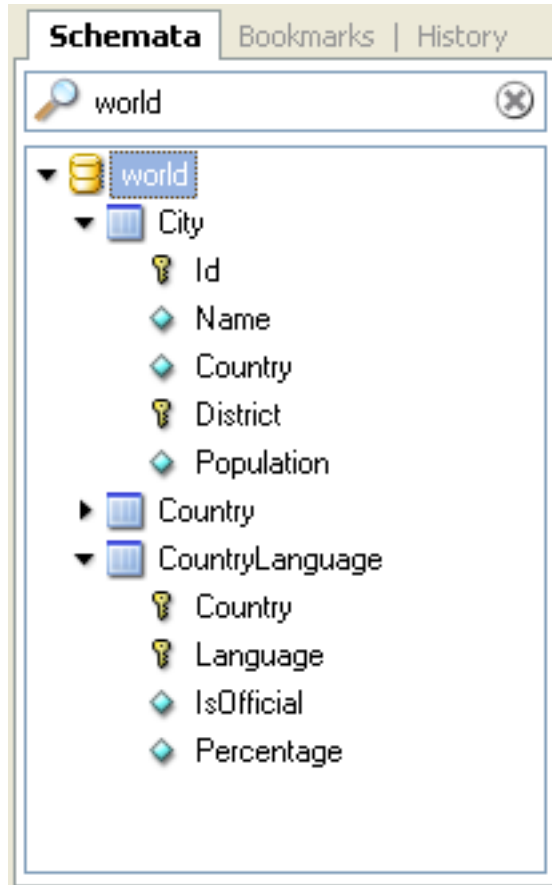
4.5. The Object Browser

The object browser allows you to browse your server's databases, your bookmarks, and your query history.

4.5.1. The Database Browser

The database (schemata) browser is the primary screen of the object browser. You can use the database browser not only to select tables and fields to query, you can also edit tables, create new tables and databases, and drop tables and databases. The database browser is also one of the places where you can set the default database, which is a required action before you can start issuing queries.

Figure 4.5. The database browser



The current default database will always be highlighted with bold text. To change the default database, right-click on the desired database and choose the Make Default Schema option.

You can filter the databases shown by using the search bar at the top of the database browser. As you fill in the search bar, databases that do not match your search are hidden from view. You can clear the search bar by clicking the X button on the right side of the search bar. All databases will once again be displayed.

To view a database's tables, stored procedures, and views, click the black arrow on the left of the database name. You can view a table's columns by clicking on the black arrow to the left of the table name. Fields that form part of an index have a small key icon to the left of their name, otherwise they have a blue diamond icon.

When you click the black arrow to the left of a view, the columns that form the view are shown. When you click the black arrow to the left of a stored procedure or stored function, the arguments of the stored procedure or function are shown.

To create a new database, right-click within the database browser and choose the Create new Schema option. To create a new table, right-click the database you wish to add a table to and choose the Create new Table option. To create a new view, right click within the database you wish to add a view to and choose the Create New View option. To create a new stored procedure or stored function, right click within the database you wish to add to and choose the Create New Procedure / Function option. You can drop objects by right-clicking on the object you wish to drop and choosing the appropriate option (Drop Schema, Drop Table, etc.).

You can edit an object by right-clicking on it and choosing the appropriate edit option (Edit Table, Edit View, etc.). When you choose the Edit Table option, the MySQL Table Editor will be displayed with the

selected table. For more information on editing tables, see The MySQL Table Editor. When the Edit View, Edit Function, or Edit Procedure options are chosen, the CREATE VIEW, CREATE PROCEDURE, or CREATE FUNCTION statement will be shown in a new tab using the Script Editor. It is possible to edit all server functions/procedures at once by choosing the Edit All Stored Procedures / Functions option from the Script menu.

4.5.2. The Bookmark Browser

You can place your more commonly used queries in bookmarks so that you can quickly retrieve them and re-use them later. To add a query to your bookmarks highlight and drag it from the query area into the bookmark browser.

Your bookmarks can be organized into folders and subfolders to help with management of your queries. To add a new subfolder right-click on an existing folder and choose the Create Bookmark Folder option. You can remove bookmarks and folders by right-clicking on them and choosing the Delete Items option. After you confirm that you wish to delete the item it will be removed from your bookmark list. *If you delete a folder all items and subfolders within the folder are also deleted.*

4.5.3. The History Browser

With the history browser you can browse through all the queries you have previously issued. To expand a given day's queries, double-click on the day. To load a history item into the query area, double-click on it or drag it to the query area.

You can remove history items by right-clicking on them and choosing the Delete Selected History Entries option from the drop-down menu. You can also use the Clear History option to erase all history entries.

You can create bookmarks from history items by right-clicking on a selected history item and choosing the Add History Item as Bookmark menu option.

4.6. The Information Browser

The Information Browser provides access to all information that is not directly related to actual data within your database. The parameter browser provides different local, global, and dynamic parameters that can help build your queries, whereas the syntax browser provides a convenient reference to MySQL query syntax. The Function browser is a quick reference to the various functions built into MySQL. The Transaction Browser lists all queries that make up a single transaction and serves as a history for a single transaction.

4.6.1. The Parameter Browser

The parameter browser contains all the local, global, and dynamic parameters that can be included in your query. At the moment only the dynamic parameters are available, which allow you to create master-detail queries.

4.6.2. The Syntax Browser

The syntax browser provides a quick reference tool for determining proper syntax to use when constructing queries. By double-clicking on the various types of syntax (SELECT, REPLACE, and so forth) you can bring up reference information using the in-line help system.

4.6.3. The Function browser

The function browser provides a quick reference tool for usage of the built-in functions of the MySQL server. By double-clicking on the various functions you can bring up reference information using the inline help system.

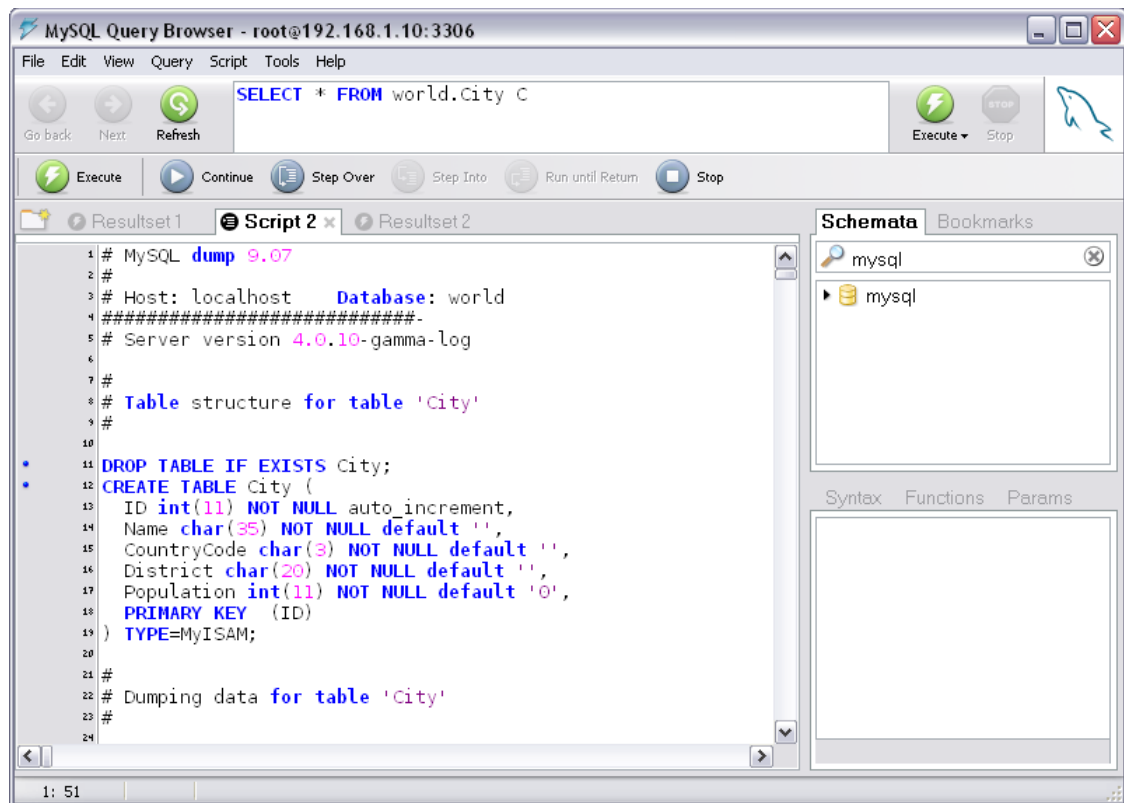
4.6.4. The Transaction Browser

The Transaction Browser provides a list of all statements that have been executed in the current transaction. The Transaction browser will be displayed when you BEGIN a transaction using the transaction control buttons.

4.7. The Script Editor

The Script Editor provides an alternate interface for use when dealing with large SQL scripts that involve multiple SQL statements.

Figure 4.6. The script editor



The Script Editor displays within an individual tab in the Result Area. When a Script Editor tab is active, the Advanced Toolbar buttons are replaced with script debugging buttons.

To open a script in the Script Editor, choose the Open Script ... option from the File menu. You can also press the **Ctrl + O** keys to open a SQL script.

To create a new Script Editor window without loading a script, choose the New Script Tab option from the File menu.

4.7.1. The Script Debugging Buttons

The following buttons are available when using the Script Editor:

- **Execute:** Execute the script from the beginning and do not stop execution until the end of the script is reached.
- **Continue:** Execute the script from the beginning (or current position) and stop for errors or breakpoints.
- **Step Over:** Execute the next statement, functions will be executed but not descended into.
- **Step Into:** Execute the next statement, functions will be descended into.
- **Run Until Return:** Run the script until the current function returns.
- **Stop:** Stop execution of the script.

4.7.2. The Script Area

The current script being edited in the Script Editor is displayed in the `Script Area`. The Script Area features line numbering and syntax highlighting to make it easier to work with scripts.

Script lines that are marked with a blue dot to the left of the line are eligible for use as breakpoints. To set a line as a breakpoint, click the blue dot. You can also set a breakpoint by positioning the cursor in the line you wish to break at and clicking the **Toggle Breakpoint** option in the Script menu. To remove all breakpoints, choose the **Remove All Breakpoints** option of the Script menu.

Chapter 5. Using The MySQL Query Browser

In this chapter we further describe the tools available in the MySQL Query Browser through practical examples of their use.

5.1. Entering and Editing Queries Manually

The most common task performed with the query browser is that of executing queries and analyzing their results. The most direct way to create a query is to type it directly into the query area. As you type in your query, the SQL syntax portions of the query(SELECT, FROM, WHERE, and so on.) are highlighted in blue and changed to uppercase.

As you enter your query, the query area will expand from an initial three lines in height to a maximum ten lines in height. For additional space, you can press the **F11** key to maximize the query area. You can also choose the Maximize Query Area of the View menu to maximize the query area. When the query is maximized, line numbers are displayed for the query, and the query area can be resized by clicking and dragging the line that divides the query area from the result area. To restore the query area, press the **F11** key again.

Once you have entered your query, click on the Execute button and your query results are displayed in the result area. You can also press **Ctrl+Enter** to execute the query. If there is an error with your query an error area appears at the bottom of the result area and displays the relevant error message and error number.

In addition to loading the query results into the current active result area, you can also create a new result area for the results of your query or split your current result area and load results into the new section.

To execute the query and load the results into a new result area click the down-arrow below the Execute button and choose the Execute in new Tab option or press **Ctrl+Shift+Enter**.

To split the active result area and display the query results click the down-arrow below the Execute button and choose the Split Tab and Execute option or press **Ctrl+Alt+Enter**.

You must set a default database before you can query the database successfully. You can set the default database at the connection screen, by right-clicking on a database in the database browser and choosing Make Default Schema, or by choosing the Change Default Schema option from the File menu.

5.2. Building Queries

One feature of the MySQL Query Browser is the ability to build queries. This allows you to select the columns and tables you wish to query from the database browser and have the query created automatically based on your choices.

The first step to building a query is to choose a table to query. Click and drag the table you wish to query to the query area to start a query. For example, by dragging the City table to the query area, `SELECT * FROM City C` is added to the query area. You can also double-click a table to start a new SELECT query.

You can drag additional tables to the query area while holding the **Ctrl** or **Shift** keys to add them to the FROM clause. By holding the **Shift** key while dragging, the tables are joined. By holding the **Ctrl** key while dragging, a `LEFT OUTER JOIN` is performed. The tables are joined based on identical column

names for MyISAM tables and foreign key information for InnoDB tables.

Figure 5.1. The table tool



When you select a table from the database browser and drag it over the query area, a table tool with some query composition actions is displayed. Drop the table you're dragging on the desired action and the query is modified accordingly. The following actions are possible:

- **SELECT** replaces the current statement with a **SELECT** query containing the dragged table.
- **Add Table** adds the dragged table to the list of tables in the current **SELECT** query
- **JOIN Table**: If a **SELECT** query is already in the query box, with a table in it, the new dragged table is added and the appropriate **WHERE** clauses to perform a **JOIN** will be added
- **LEFT OUTER JOIN** is the same as the previous, but does a **LEFT OUTER JOIN** instead of a **JOIN**
- **UPDATE** replaces the current statement with an **UPDATE** statement containing the dragged table
- **INSERT** replaces the current statement with an **INSERT** statement containing the dragged table
- **DELETE** replaces the current statement with a **DELETE** statement containing the dragged table

Once a table is selected, you can choose specific columns to query; click the **Select** button from the query building buttons on the button bar. Your mouse pointer is changed to a **Select** pointer, which you can use to choose columns from the database browser. For example, by clicking on the **Id**, **Name**, and **Country** fields of the **world** sample database, the query **SELECT C.Id, C.Name, C.Country FROM City C** is built in the query area.

Once you have chosen the columns you wish to query, you can use the other query building buttons to complete your query with **WHERE**, **GROUP BY**, and **ORDER BY** clauses. When a new section of the query is added with the query building buttons, the cursor in the query area is placed in position for editing; if you click a field with the **WHERE** pointer, the cursor is in position for you to type in the details of the **WHERE** clause.

You can change between the different query building pointers by clicking on the query building buttons in the button bar, or by using a combination of **Ctrl+Alt** and the first letter of the pointer you wish to use (**Ctrl+Alt+S** for **SELECT**, **Ctrl+Alt+W** for **WHERE**, and so forth.)

5.3. Creating Views

The simplest way to create a view using the MySQL Query Browser is to use the **Create View** button. Execute a query that represents the view you wish to create. Once the query is executed, click the **Create View** button, provide a name for the view, and the view is created.

You can also create a view by right-clicking on the database you wish to add a view to in the database browser and choosing the **Create New View** option. After you enter a name for the view, a view template will be displayed in the script editor.

5.4. Navigating Result Sets Within the Result Area

Once you have successfully executed a query you are then able to view and manipulate the result set within the result area.

You can navigate the result area using the arrow keys, tab key, and PageUp/PageDown keys. The Home and End keys can be used to move to the first and last column within a given row. Your current position within the result set is shown in the bottom-left corner of the application window. The First and Last buttons at the bottom of the result area can be used to move to the first and last rows of the result set.

NULL data will be indicated with a special NULL icon to differentiate NULL data from empty strings. BLOB fields will be empty and will have a special BLOB icon.

When navigating a particularly large result set, you can press the **F12** key to maximize the result area. You can also choose the Maximize Tabsheets option of the View menu to maximize the result area. Pressing **F12** again will restore the result area to its original size.

To search for a particular value within the result set, click the Search button. To see additional search options, click the Details >> button. The following options are available:

- **Case Sensitive:** The search is performed in a case-sensitive manner. By default, searches are not case-sensitive.
- **Whole Words Only:** The search does not allow partial matches. By default, partial matches are allowed (i.e. Edm will match Edmonton).
- **Search From Top:** The search begins with the first row in the result set.
- **Search From Cursor:** The search begins from the currently selected row.
- **Search all Text / Columns:** The search will involve all columns of the result set.
- **Search Only in Selected Text / Column:** The search will only involve the currently selected column(s) of the result set.
- **Search Up:** The search will scan the result set, starting at the designated start location and moving upwards.
- **Search Down:** The search will scan the result set, starting at the designated start location and moving down.

To perform a search and replace operation on a result set, click the Replace tab in the search dialog to enter search and replace mode. The options for performing a replace operation are the same as for a search. To replace a single instance of a string, click the Replace button. To replace all instances of a string, click the Replace All button.

5.5. Exporting Result Sets

You can export any result set from MySQL Query Browser by right-clicking within the result set and choosing an option from the Export Resultset sub-menu. You can choose to export the result set in CSV, XML, HTML, or Microsoft Excel XLS formats.

5.6. Handling BLOB and TEXT Columns

The MySQL Query Browser provides functionality for dealing with BLOB and TEXT columns through a series of special icons.

Figure 5.2. The BLOB handling icons



These icons appear in any BLOB or TEXT columns in your result set. From left to right the following icons are available:

- **Open File:** This icon looks like a file folder and is used to open a file for loading into the field.
- **View:** This icon looks like an eye and is used to open the field viewer to view the contents of the field. The field viewer can be used to view TEXT fields and BLOB fields that contain images.
- **Edit:** This field looks like a wrench and opens the field viewer in edit mode, allowing you to make changes to the data and apply the changes to the result set.
- **Save:** This icon looks like a floppy disk and is used to save the contents of a TEXT or BLOB field into a file.
- **Clear:** This icon looks like a red X within a red box and is used to clear the contents of a TEXT or BLOB field.

Only the View and Save icons are visible if you have not enabled editing for your record set. See Section 5.7, “Editing Result Sets Within the Result Area” for information on editing result sets.

5.7. Editing Result Sets Within the Result Area

When a query is based on a single table, with sufficient identifying information, the result set can be edited from within the result area. To edit a result set click the Edit button. If the Edit button is not active, your result set is not editable.

Once in edit mode you can insert, update, and delete rows from the result set visually. You can navigate the fields with the **Tab** and arrow keys, and pressing **Enter** allows you to edit the content of a field. You can also double-click a field to make it editable. When editing a field, the tab key will move you to the next field in an editable state. All edited fields are highlighted in blue for easy identification.

To add rows to the result set, scroll to the bottom of the result area and fill in the fields of the blank row found there. All new rows are highlighted in green.

To delete a row right-click on the row and choose the Delete Row option. All deleted rows are highlighted in red.

Changes made to the result set are not applied immediately, but instead are cached until the Apply Changes button is pressed. You can abort your edits with the Discard Changes button. Exiting edit mode without choosing to apply or discard your changes results in you being prompted to apply or discard your work.

5.8. Comparing Result Sets

You can compare result sets graphically with the MySQL Query Browser, allowing you to easily determine where rows have been inserted, updated, or deleted.

To compare two result sets, execute the first of the queries you wish to compare. Once the result set has loaded, right-click on the result set and choose the Split Tab Horizontally option. Load your second query into the new section of the result area and click the Compare button to compare the two result sets.

When you have activated the compare mode, both result sets will scroll in unison, both vertically and horizontally. Rows are matched for comparison, with blank rows added when one set has a row that the other set lacks.

If one result set has a row that the other result set does not have, that row is highlighted in green. The other result set has a blank row inserted that is highlighted in red. If both result sets have a matching row, but individual fields are different, those fields are highlighted in blue.

In order to successfully compare two result sets, you need two queries with matching column names. The tables which the data is based on need to have primary keys defined in order for the MySQL Query Browser to match rows. Fields in your result sets must have the same names, and must be in the same order.

5.9. Creating Master-Detail Views

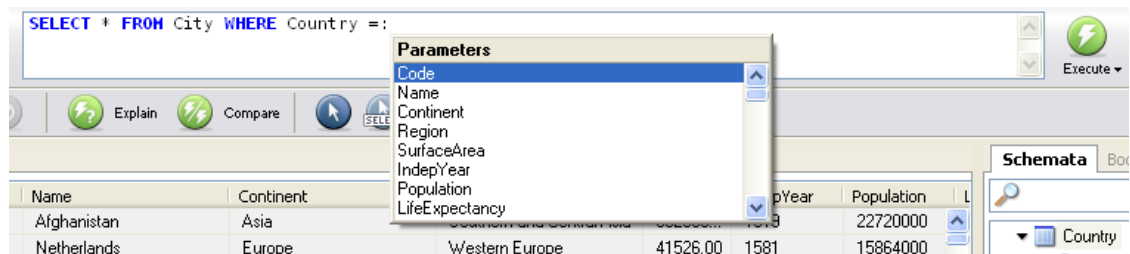
MySQL Query Browser makes it easy to view data that is in a master-detail (one to many) relationship. This can be useful for viewing customer/order data, group/member data, and so forth.

To display a master-detail view, first query the master table in a new result area. For example, using the world sample database, you could issue a query like `SELECT Code, Name, Continent FROM Country`.

Once you have created the master result set, right-click on the result area and choose the Split Tab Vertically option. For the detail query, create a query that uses the dynamic parameters available from the master query, as seen in the parameter browser. In this example, we could query the City table like this: `SELECT Id, Name FROM City WHERE Country = :Code`.

The colon character indicates to the MySQL Query Browser that you are adding a dynamic parameter, and a pop-up list of available parameters should appear within the query area, allowing you to choose a parameter using the arrow keys. You can also type the name of the parameter that you would like to use to link the detail query.

Figure 5.3. The dynamic parameter list



Once you execute the detail query it will automatically refresh any time you change the active row in the master query, allowing you to quickly see all detail rows as you navigate the master result set.

You can repeat this process, splitting the result area additional times, allowing you to have detail result sets for detail result sets.

5.10. Editing Queries From a Development Tool

In order to help programmers optimize and troubleshoot their queries more efficiently, the MySQL Query Browser can copy queries from application code using your favorite IDE.

This functionality is only available for the Windows version of MySQL Query Browser.

The following PHP code will be used as an example:

```
$SQL = "SELECT Id, Name, Country FROM City";  
mysql_query($SQL,$db);
```

To copy the listing into the MySQL Query Browser, highlight the query from within your editor and press **Ctrl + Alt + Shift + C**. The non-query portions of the highlighted area will be stripped and the query will be pasted into the query area.

After editing the query, press **Ctrl + Alt + Shift + V** to load the modified query back into your original code. The surrounding PHP code will be re-inserted along with the modified query. This functionality allows you to edit queries quickly while programming.

In order to use this functionality your editing application needs to support the standard Windows clipboard system, which most Windows applications do.

Chapter 6. The MySQL Table Editor

6.1. Introduction

The MySQL Table Editor is a component of both the MySQL Query Browser and MySQL Administrator, and allows for the visual creation and modification of tables.

The MySQL Table Editor can be accessed from the MySQL Query Browser by right-clicking on a table within the database browser and choosing the Edit Table option, or by right-clicking on a database within the database browser and choosing the Create New Table option.

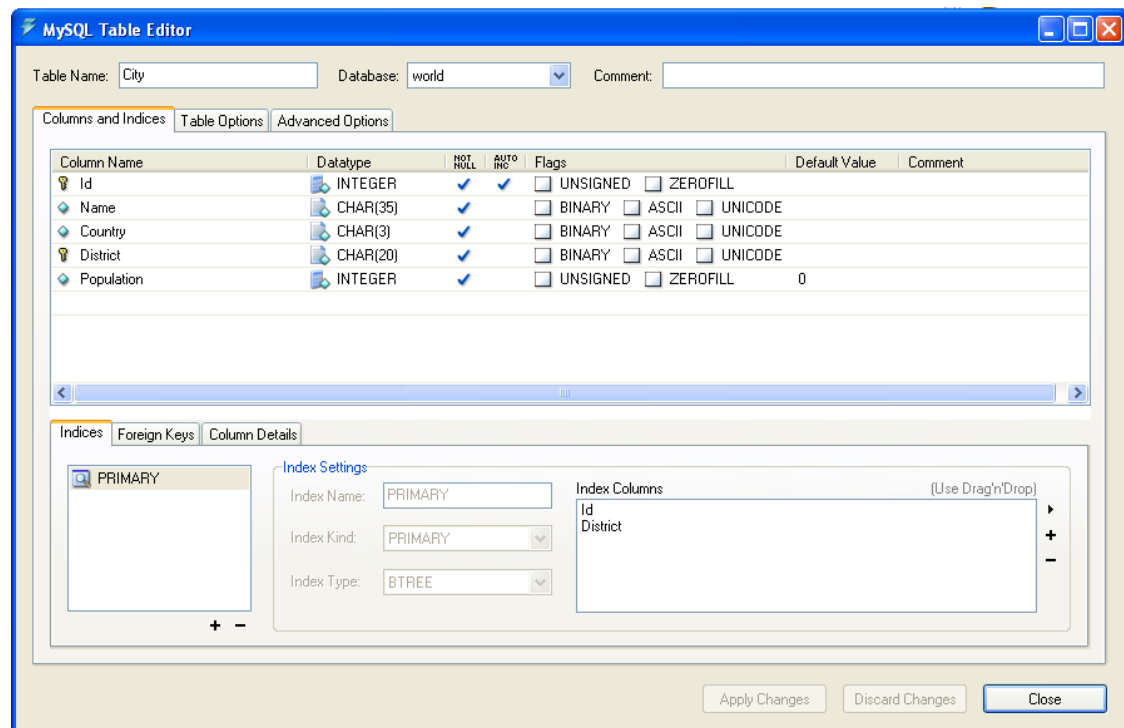
The MySQL Table Editor can be accessed from MySQL Administrator through the Catalogs screen. Once you have selected a database, right-click on a table and choose the Edit Table option from the drop-down menu. You can also select a table and click the Edit Table button to access the MySQL Table Editor.

6.2. The Main Editor Window

The MySQL Table Editor consists of a work space divided into three tabs, some general information prompts, and three action buttons.

Regardless of the tab you have active, you can always edit the table name, the database it belongs to, and the table comment.

Figure 6.1. The MySQL Table Editor



The tabbed area is divided into three sections:

- *Columns and Indices:* Use the Columns and Indices tab to create and modify the table's column and index information. You can also create FOREIGN KEY relationships using this tab.
- *Table Options:* Use the Table Options tab to choose the storage engine used by the table and the table's default character set.
- *Advanced Options:* Use the Advanced Options tab to configure advanced options such as per-table storage directory, MERGE and RAID table options, and table/row length options.

Each of these areas are discussed in further detail in the following sections.











6.3. The Columns And Indices Tab

The Columns and Indices tab can be used to display and edit all column and index information for your table. Using this tab, you can add, drop, and alter columns and indexes.

6.3.1. The Column Editor

You can use the column editor to change the name, data type, default value, and other properties of your table's columns.

Figure 6.2. The column editor

Column Name	Datatype	NOT NULL	AUTO INC	Flags	Default Value	Comment
 Id	 INTEGER	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL		
 Name	 CHAR(35)	<input checked="" type="checkbox"/>		<input type="checkbox"/> BINARY <input type="checkbox"/> ASCII <input type="checkbox"/> UNIC		
 Country	 CHAR(3)	<input checked="" type="checkbox"/>		<input type="checkbox"/> BINARY <input type="checkbox"/> ASCII <input type="checkbox"/> UNIC		
 District	 CHAR(20)	<input checked="" type="checkbox"/>		<input type="checkbox"/> BINARY <input type="checkbox"/> ASCII <input type="checkbox"/> UNIC		
 Population	 INTEGER	<input checked="" type="checkbox"/>		<input type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL	0	

To change the name, data type, default value, or comment of a column, double-click on the value you wish to change. The value becomes editable and you can complete your changes by pressing the **Enter** key.

To modify the flags on a column (UNSIGNED, BINARY, ASCII, and so on) check and uncheck the boxes corresponding to the flag you wish to change.

To the left of the column name is an icon that indicates whether the column is a member of the primary key. If the icon is a small key, that column belongs to the primary key, otherwise the icon is a blue diamond. To add or remove a column from the primary key, you can click on the icon.

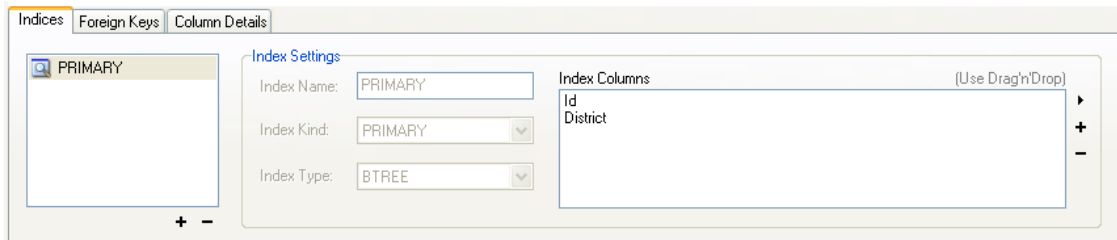
6.3.2. The Detail Area

The detail area of the Columns and Indices tab is divided into three tabs and is used for modifying the index, foreign key, and miscellaneous information regarding your columns.

6.3.2.1. The Indices Tab

The `Indices` tab holds all index information for your table. You can add, drop, and modify indexes using the indices tab, in combination with the column editor.

Figure 6.3. The index editor



To add an index, click the + (plus) button below the index list. You will be prompted for an index name and then the new index is created. To drop an index, select the index and click the - (minus) button to drop the index from the index list.

Use the Index Name and Index Kind dialogs to modify the name and type (`UNIQUE`, `FULLTEXT`, and so on) of the index.

To add columns to an index, either click and drag the column to the Index Columns box or select the column you wish to add and click the + (plus) button to the right of the Index Columns box. You can remove a column from the index by selecting the column and clicking the - (minus) button to drop the column from the index.

To change an index to only refer to a column prefix (such as with the `CHAR` and `VARCHAR` string data types), select the index column you wish to prefix in the Index Columns box and then click the arrow icon to the right of the Index Columns box. Select the Set Index Column Length option from the drop-down menu that appears. You can set the desired prefix length in the dialog that appears.

6.3.2.2. The Foreign Keys Tab

The `Foreign Keys` tab is divided into two sections, one with a list of foreign keys and one with various dialogs for foreign key settings.

To add a foreign key, click the + (plus) button below the foreign key list. You will be prompted to name the new foreign key, and it is then be added to the list.

To drop a foreign key, select the foreign key and click the - (minus) button below the foreign key list. The foreign key is then dropped from the foreign key list.

You can modify the name of the foreign key, its `ON DELETE`, and its `ON UPDATE` actions using the dialogs provided in the Foreign Key Settings section of the tab.

To establish a foreign key relationship, choose a table from the Ref. Table drop-down list. The columns that can be referenced as foreign keys are listed in the area below, and you can drag a column from the column editor to the Column section to the left of the column you wish to reference.

6.3.2.3. The Column Details Tab

The `Column Details` tab provides an interface for setting the parameters of a column without using the table interface of the column editor.

All settings that are available in the Column Editor are also available in the Column Details tab, and in addition you can also configure the column character set and column default collation from the Column Details tab.

6.4. The Table Options Tab

The `Table Options` tab allows you to change the storage engine and character set of your table. The potential storage engines are listed, along with a brief summary of each storage engine's features and strengths.

To change the storage engine for your table, click on the radio button next to the desired storage engine.

To change the default character set or collation of your table, choose a new option from the drop-down list of available character sets.

6.5. The Advanced Options Tab

The `Advanced Options` tab is used to configure table options that would be considered outside the standard set of options that most users designate when creating and modifying tables.

The `Advanced Options` tab is divided into several sub-sections, each of which is described in the upcoming sections of this manual.

6.5.1. The Various Section

The `Various` section of the `Advanced Options` tab contains options for you to set the `PACK KEYS` behavior, the table password, the initial `AUTO_INCREMENT` value, and the delayed key update behavior.

The `AUTO_INCREMENT` and delayed key update behaviors apply only to MyISAM tables.

6.5.2. The Row Options Section

The `Row Options` section can be used to configure options such as the row format, checksum use, and the row size parameters needed for large tables.

To set the row format, choose the desired row format from the drop-down list. See http://dev.mysql.com/doc/mysql/en/MyISAM_table_formats.html for more information on the different row formats that are available. This option only applies to MyISAM tables.

When you expect a table to be particularly large, use the `Avg Row Length`, `Min Rows`, and `Max Rows` options to enable the MySQL server to better accommodate your data. See http://dev.mysql.com/doc/mysql/en/CREATE_TABLE.html for more information on how to use these options.

6.5.3. The Storage Options Section

The `Storage Options` section is used to configure a custom path to the table storage and data files. This option can help improve data integrity and server performance by locating different tables on different hard-drives.

This option is only available for MyISAM tables and is not available for servers running under the Windows operating system.

6.5.4. The Merge Table Options Section

The Merge Table Options section is used to configure MERGE tables in MyISAM. To create a MERGE table, select MERGE as your storage engine in the Table Options Tab and then specify the tables you wish to MERGE in the Union Tables dialog.

You can also specify the action the server should take when users attempt to perform INSERT statements on the merge table. See <http://dev.mysql.com/doc/mysql/en/MERGE.html> for more information on MERGE tables. [<http://dev.mysql.com/doc/mysql/en/MERGE.html>]

6.5.5. The Table RAID Settings Section

The Table RAID Settings section allows you to configure RAID support for MyISAM tables. RAID allows MyISAM table data files to grow larger than the 2GB/4GB size limit imposed by some operating systems.

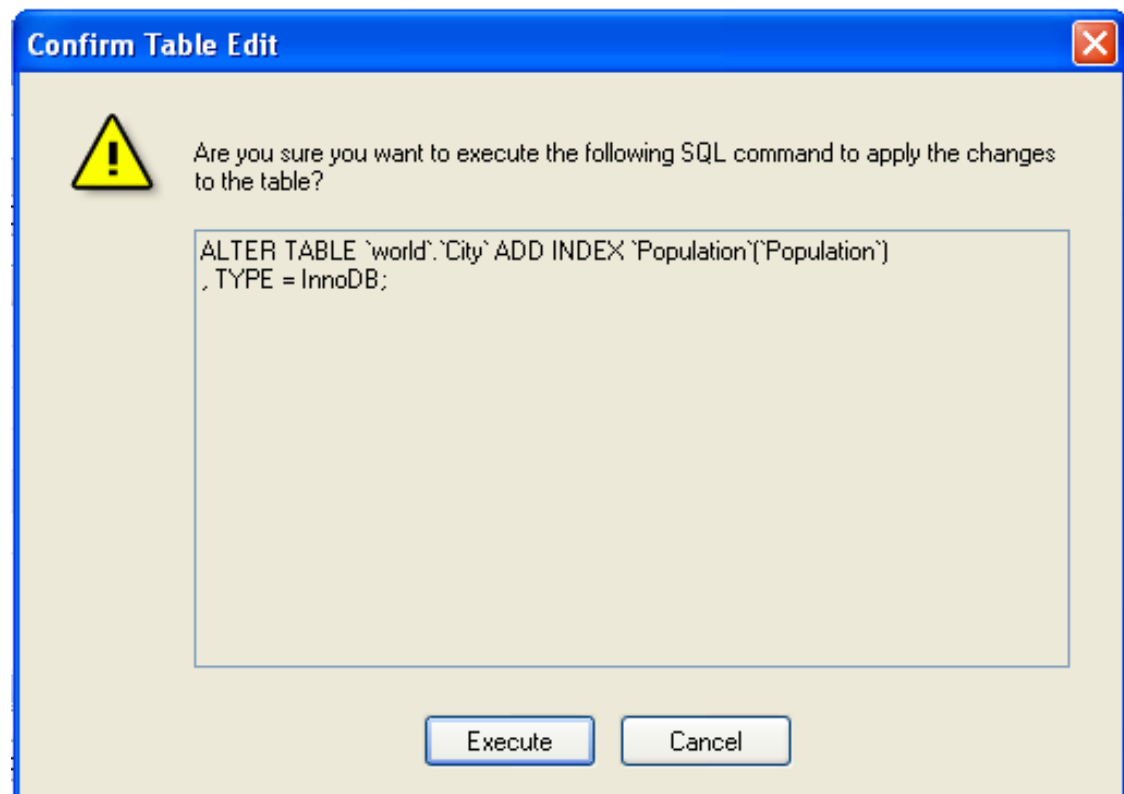
For more information on using RAID support with MyISAM, see http://dev.mysql.com/doc/mysql/en/CREATE_TABLE.html

6.6. Applying Your Changes

The changes you make with the MySQL Table Editor are not immediately applied but are instead queued to be applied in batches after you have made all your edits.

To apply the changes you have made, click the Apply Changes button. The Confirm Table Edit dialog will appear.

Figure 6.4. The Confirm Table Edit dialog



You can click the Execute button to confirm the changes and have them applied, or click the Cancel button to discard the changes. You can also click the Discard Changes button in the main MySQL Table Editor window to discard all changes you have made.

You can also copy the proposed changes to the clipboard for further editing by highlighting the ALTER TABLE or CREATE TABLE statement, right-clicking and choosing Copy from the drop-down menu.

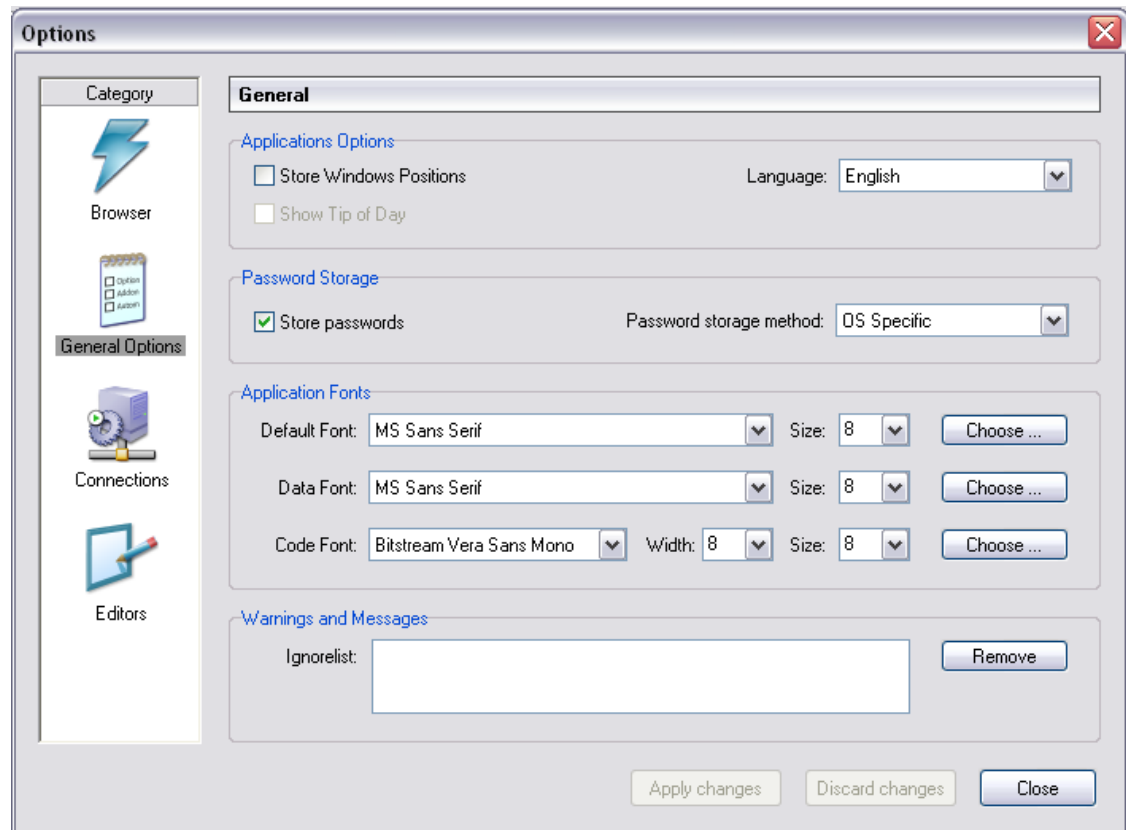
Chapter 7. Options Dialog

7.1. Introduction

The Options dialog allows you to configure connection profiles, general program settings, and more. You can open the Options dialog window using one of the following methods:

- In the connection dialog window, click the ... button.
- In the main application window select Options ... from the Tools menu.

Figure 7.1. Options dialog



In the sidebar of that dialog, you can select the section you wish to configure. There are three action buttons in the lower right corner of the window:

- Apply Changes applies and saves changes.
- Discard Changes discards any changes you have made.
- Close closes the Options dialog window. If you have not applied or discarded your changes, you will be prompted to do so.

7.2. General Options

The `General Options` section allows you to specify a number of settings that are valid for all graphical MySQL applications. At the time of writing, MySQL Administrator and MySQL Query Browser are the only programs in the MySQL GUI suite, but other programs will follow soon.

- `Store Window Positions`: When an application is started the next time, its recent window positions will be reused.
- `Show Tip of Day`: If checked, a pop-up window with the tip of the day appears at program startup.
- `Store Passwords`: If checked, passwords are stored in the user's connection profile. In this case, you can select the password storage method:

`Plaintext` means they are stored without encryption; this can be insecure. `Obscured` means they are encrypted using a weak algorithm. This encryption method, however, is operating system independent. Selecting `OS Specific` will use the default encryption method provided by your operating system. The default option is `Plaintext`.

- `Language`: Select the interface language. The default is `English`.
- `Default Font`: The font used for all application text.
- `Data Font`: The font used for all query and table data displayed.
- `Code Font`: The font used for all queries entered by the user..
- `Ignorelist`: Whenever you check the `Do Not Show This Message Again` option on error and message prompts, they are added to this list. If you would like a particular message to be shown again, click the `Remove` button after selecting the message from the list.

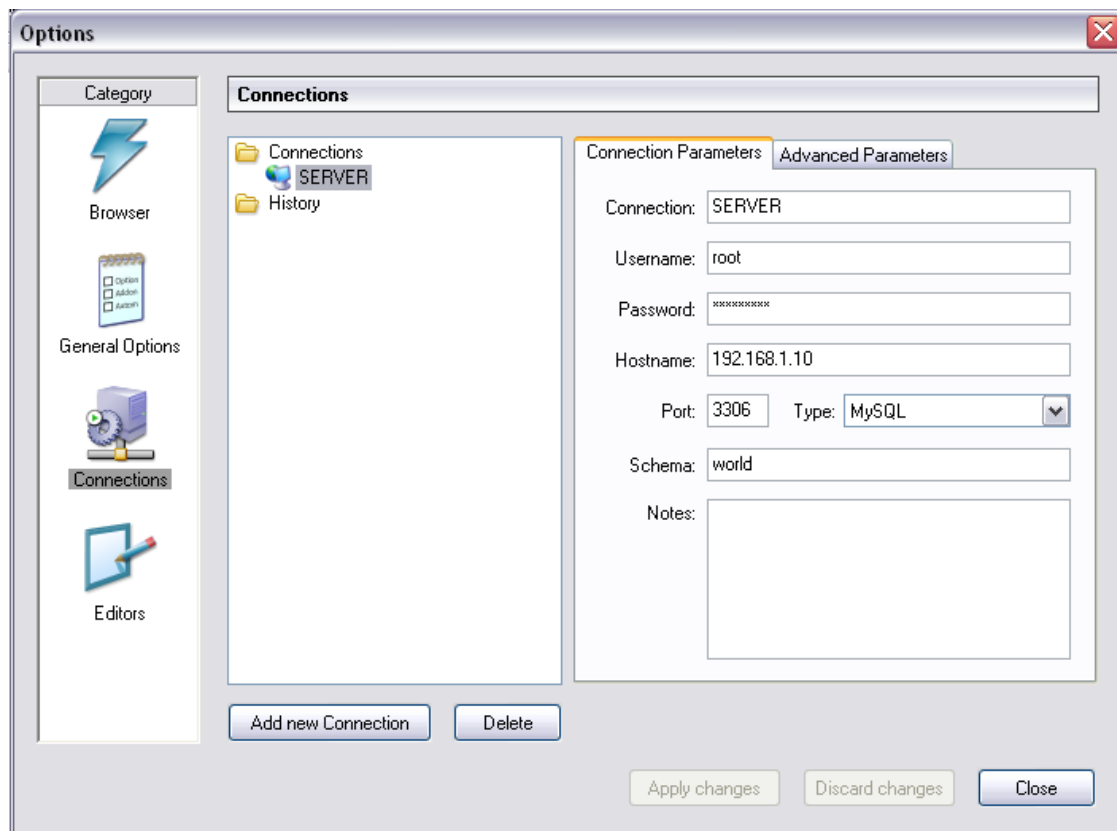
You may change the font and font size of any of the application fonts by clicking the `Choose ...` button to the right of the font.

7.3. Connections

The `Connections` section allows you to create, edit, and delete connection profiles. The center box displays a list of currently available profiles, together with a history of connections that were made without being stored in a profile. You can collapse or expand both the `Connections` and `History` trees by double-clicking them.

Connections are automatically added to the `History` tree whenever you establish a connection to a MySQL server without using one of the profiles stored under the `Connections` tree. They do not appear in the drop-down box of the `Connection` dialog, but you can use any of them by manually typing their name into the `Connection` box of the `Connection` dialog.

Figure 7.2. Options dialog: Connections



To edit an existing connection profile, click on its name and change the values that appear in the Connection Parameters and Advanced Parameters tabs, then click on the Apply Changes button to save your changes.

When you select a connection profile from either the Connections or History trees, the Connection Parameters tab displays the following fields:

- **Connection:** The connection profile label. This is the name by which you refer to the profile and that appears in the Connection drop-down box of the Connection dialog. It may contain any characters, including spaces. Choose distinctive names so that you can easily tell which profiles they refer to. The names can help you distinguish connections to different MySQL servers, or connections as different MySQL users to a given server.
- **Username:** The username used to connect to the MySQL server.
- **Password:** The password used to connect to the MySQL server. Note that passwords are not stored in the connection profile, unless you specify otherwise in the General Options section.
- **Hostname:** The name of the host machine where the MySQL server runs, or its IP address.
- **Port:** The TCP/IP port that the MySQL server listens to on the host machine.
- **Type:** Specifies the protocol used to connect to the database server. The default protocol is MySQL (which uses the native MySQL protocol). The other protocols listed are not currently available.
- **Schema:** The default database for a connection when using the MySQL Query Browser.
- **Notes:** You can use this field to enter comments or additional information describing the connection.

tion profile.

Note that the Advanced Parameters tab may not be available in some of the first releases of MySQL Administrator. If that is the case in the version you are using, you can still set those parameters in the Connection dialog. Use the Details ... button of that dialog to display the Advanced Connection Options.

When you select a connection profile from either the Connections or History list, the Advanced Parameters tab displays the following checkboxes:

- `Use compressed protocol`: If checked, the communication between the application and the MySQL server will be compressed, which may increase transfer rates. This corresponds to starting a MySQL command-line tool with the `--compress` option.
- `Return number of found rows, not number of affected rows`: By default, MySQL returns the number of rows changed by the last `UPDATE`, deleted by the last `DELETE` or inserted by the last `INSERT` statement. When this option is checked, the server returns the number of rows matched by the `WHERE` statement for `UPDATE` statements.
- `Ignore spaces after function names, make them reserved words`: Normally, any reference to a function name in an SQL statement must be followed immediately by an opening parenthesis. If this option is checked, spaces may appear between the function name and the parenthesis, like this:

```
COUNT ( * )
```

Enabling this option has the effect that function names become reserved words. This option corresponds to starting a MySQL command-line tool with the `--ignore-spaces` option.

- `Allow interactive_timeout seconds of inactivity before disconnect`: Normally, the connection is closed by the MySQL server after a certain period of inactivity on the client side. This period can be set with the `interactive_timeout` variable. If checked, the server will not close the connection unless the period of inactivity exceeds the value set by `interactive_timeout`. This corresponds to starting a MySQL command-line tool with the `--connect-timeout=seconds` option.
- `Enable LOAD DATA LOCAL handling`: By default, the `LOCAL` option of the `LOAD DATA` statement is disabled for security reasons. Enabling this option will allow to load data from the local machine (the machine where the client GUI application is running). This option corresponds to starting a MySQL command-line tool with the `--local-infile=1` option. (Note that this option is ineffective unless the MySQL server allows `LOCAL` handling.)

7.4. Editors

The Editors section is used to configure options specific to the different editors available within the MySQL GUI Suite. At the time of writing this is limited to the MySQL Table Editor.

- `Show SQL command before applying changes` toggles whether the MySQL Table Editor will show you the `ALTER TABLE` statement it is about to execute for confirmation when you click Apply Changes.
- `All columns Not Null per default` determines whether the MySQL Table Editor will designate columns as being `NOT NULL` by default when creating new columns.

- `All integer columns unsigned per default` sets whether integer columns are declared `UNSIGNED` by default when creating new columns.
- `PK Naming` dictates the format that should be used to name `PRIMARY KEY` columns that are automatically generated. the `%tablename%` portion will be replaced with the name of the appropriate table.
- `Index Naming` sets the name automatically generated for new indexes. The `%nr%` string will be replaced with an automatically incrementing number.
- `FK Naming` configures the name used when creating new foreign keys.
- `PK Datatype` is the datatype used when creating a `PRIMARY KEY` column.
- `Def. Datatype` is the default datatype assigned to all new columns that are not part of a `PRIMARY KEY`.

7.5. The Browser Section

The Browser section allows you to specify options that relate to the MySQL Query Browser. The Browser section is divided into three areas: Display Options, Query Options, and Various.

7.5.1. Display Options

The display options affect the appearance of the MySQL Query Browser.

`Show advanced toolbars` turns on and off the display of the The Advanced Toolbar.

`Hide the tab when only one tab is open` determines whether or not to display a tab icon at the top of the Result Area if there is only one active result area.

`Toolbars use gradient background` enables and disables use of gradient effects in the background of the Query Toolbar.

`Show field overlay images for long VARCHAR, TEXT, and BLOB fields` toggles on and off the BLOB management icons.

7.5.2. Query Options

`Enforce queries to be editable by adding primary key columns to the select` will add `PRIMARY KEY` columns to a query that does not explicitly `SELECT` them to ensure that the results of a query can be edited. The `PRIMARY KEY` columns will not be displayed in this case, but will still be stored.

`Open resultset in associated application after export` will cause the associated application to be opened after choosing the `Expost Resultset` option from the right-click menu of the Result Area.

7.5.3. Various

`Associate sql/query files with Query Browser` will configure Windows to associate all `.sql` files with the MySQL Query Browser. As a result, all such files will be opened with MySQL Query Browser by default.

Appendix A. Installing From Source

A.1. Introduction

Normally, you will want to install the MySQL GUI tools from binaries. We put a lot of effort into making sure that our binaries are built with the best possible options. If, for whatever reason, you wish to compile the tools yourself, follow these instructions.

A.2. Downloading The Source Code

You can get the source code of the MySQL GUI tools from our public BitKeeper trees. To be able to access trees, you have to have BitKeeper installed. BitKeeper is freely available from Bitmover [<http://www.bitmover.com/>]. To run under Windows, BitKeeper requires Cygwin. If, during the installation process, BitKeeper finds that either Cygwin is not installed, or that the Cygwin version is too old, it will suggest to download and install Cygwin (or a newer version). You may, however, install Cygwin before installing BitKeeper. You can get Cygwin from Cygwin.com [<http://www.cygwin.com/>].

Our public BitKeeper trees that contain the MySQL GUI tools source code can be retrieved by the following BitKeeper commands:

```
bk clone bk://mysql.bkbits.net/mysql-administrator mysql-administrator
bk clone bk://mysql.bkbits.net/mysql-gui-common mysql-gui-common
bk clone bk://mysql.bkbits.net/mysql-query-browser mysql-query-browser
```

You will need the `mysql-gui-common` tree regardless of which GUI tools you intend to build. The `mysql-gui-common` tree contains source and graphics files used for all MySQL GUI tools.

You can also download the source for the various GUI tools from the downloads page [<http://dev.mysql.com/downloads>].

A.3. Building from Source under Windows

Installing from source under Windows requires multiple commercial compilers and because of this it is preferable to install from binaries. If you meet the prerequisites needed you can build the MySQL GUI tools from source under Windows.

A.3.1. Prerequisites

To compile the MySQL GUI tools from source under Windows, you need Borland Delphi 7, Microsoft Visual Studio.NET 2003, and some libraries which are available on the internet as open source software. Make sure the programs `DCC32.exe` and `DEVENV.exe` / ? can be launched from the command line.

You will need the following libraries to build from source:

- glib-2.0
- libxml-2
- mysql client libraries (4.0 or newer, 5.0 preferred)
- pcre-1.4

- MS Visual C runtime libraries 7.1
- Java include files 1.4.2_04
- Lua 5.0.2

A.3.2. Compiling and Building

Make sure you have cloned the `mysql-administrator`, `mysql-gui-common`, and `mysql-query-browser` trees, and that their local directories are located in the same directory, for example in `Personal Files\MySQLGUI\`.

The source files and libraries should be placed into the following tree structure:

```
Working Directory
|
- mysql-administrator
- mysql-gui-common
- mysql-query-browser
- mysql-gui-win-res
  |
  -include
  |
  | -glib-2.0
  | -java
  |   |
  |   -win32
  |
  -libxml
  -lua
  -mysql
  -pcre
  -windows
  |
  -lib
  |
  | -glib-2.0
  | -java
  | -libxml
  | -lua
  | -mysql
  | -pcre
  | -windows
```

A.3.2.1. Building from Command Line

To build from the command line open a windows command line prompt and change into the main repository of the tool (i.e. for MySQL Administrator this is `work\mysql-administrator`). There are three batch-files to start the build:

- `build_debug.cmd`
- `build_release.cmd`
- `build_commercial.cmd`

Execute one of those to generate a runtime image of the application. It will be generated in a

bin\windows directory.

Another method is to doubleclick on one of the `build_<buildtype>.cmd` files, where *buildtype* is one of debug, release, and commercial. The commercial build script is for users with commercial licenses, and makes use of the commercial version of the MySQL client library.

A.3.2.2. Building from the IDE

Before you can open any projects, you have to install these components. Otherwise some of the forms will not open and you will receive error messages.

To install the components, goto the `mysql-gui-common\source\windows\` directory. Copy the following directories to your Delphi source directory (`C:\Program Files\Borland\Delphi7\Source\`).

- png
- SynEdit
- TNT
- UniCodeEditor
- VirtualTreeView

Then doubleclick each *.dpk file outlined below. It will open Delphi. Press compile and install to install the components. Repeat this for each *.dpk file.

- SynEdit\Packages\SynEdit_D7.dpk
- TNT\Packages\TntUnicodeVcl_D70.dpk
- VirtualTreeView\Packages\VirtualTreesD7D.dpk
- UniCodeEditor\Packages\UniCodeEditorD7.dpk

These components have to be installed via the Install option of the Component menu. Click on the Browse button and select the the files below. Press OK after each file.

- png\PNGImage.pas

After installing the components you can open the projects as outlined below.

To build the application from the Delphi 7 IDE simply doubleclick the `edit_<xxxx>.cmd`. This will launch Delphi 7 and open the file. To generate the complete runtime image select Build All Projects option from the Project menu.

Like for building from command line there are three different files.

- edit_debug.cmd
- edit_release.cmd

- `edit_commercial.cmd`

The runtime image will be generated in a `bin\windows` directory.

Opening the Delphi project not using the `edit_.cmd` script will lead to incorrect project settings and problems while building the projects.*

A.4. Building from Source under Linux

A.4.1. Prerequisites

To be able to build the MySQL GUI tools from source, you need the following software:

- g++ development suite
- gtk-2.x
- gtkmm-2.0 or gtkmm-2.2
- libglade-2.x
- libxml-2.6.x
- mysql client libraries (4.0 or newer, 5.0 preferred)
- pcre-1.4
- autoconf-2.54, automake-1.7.x

In addition, you will need `gtkhtml-3.0.x` to build the MySQL Query Browser.

A.4.2. Compiling and Building

To build any of the tools, you need to have the bk tree for `mysql-gui-common` and the tree for the desired app (i.e.: `mysql-administrator`, `mysql-query-browser`) in the same top-level directory.

First build `mysql-gui-common`:

1. `cd mysql-gui-common`
2. `sh ./autogen.sh`
3. `./configure --prefix=/opt/mysql-administrator`
4. `make`
5. `make install`

The `--prefix` option can have whatever value you want, normally one would specify `/usr/local` (the default) or `/usr`, but for making precompiled binaries, something in `/opt` might be more appropriate.

After you have successfully built `mysql-gui-common`, you can build the individual tools (in this example we will build `mysql-administrator`):

1. `cd mysql-administrator`
2. `sh ./autogen.sh`
3. `./configure --prefix=/opt/mysql-administrator`
4. `make`
5. `make install`

If everything went well, things should be properly built and installed.

A.4.2.1. Building Static Binaries

Building static binaries is non-trivial under Linux, due to the enormous amount of library dependencies that come from GNOME and associated libraries. Additionally:

- Many libraries do not come with their static (`lib*.a`) versions shipped, so we need to build them ourselves.
- Some libraries can't be linked statically, such as `glibc`, because of `libnss*.so`.
- Some libraries depend on data files and modules from the original package, esp. in case of `gtk/gnome`. Problems don't just appear when the user don't have these data files installed; things may not work if they use distributions that put data files in different paths.

The current solution is to build a partially static binary, with the most common libraries left dynamically linked (`glibc` and `gtk`). `gtkmm`, `gtksourceview`, `gtkhtml` and dependencies (like `gnome`) are being linked statically. The ideal solution would be to build custom binaries for each major/chosen distribution, but we don't have time/resources for that at the moment.

To build these binaries:

1. Look at the `source/linux/static_make` script for all the files that should be removed/moved out of `/usr/lib`.
2. Run the script.

That's needed to force the linker to look for the static version of each library and also to explicitly list the dependency libs that were otherwise linked to the dynamic libs.

A.4.2.2. Building RPMs

For building RPMs, a spec file is already supplied and will be made automatically after `./configure` is ran. The spec file expects a source tarball with the following structure:

```
mysql-administrator/ mysql-administrator/mysql-gui-common/* mysql-administrator/my
```

The contents of each subdir is the entire bk tree for each tool (Windows specific files and bk metadata

are optional, of course). You must make a tar.gz file and put it in `/usr/src/redhat/SOURCES` (or whatever is your rpm SOURCES directory).

Then, execute: `rpmbuild -ba mysql-administrator.spec`

That should build the srpm and rpm files for the tool.

Appendix B. Troubleshooting Application Errors

B.1. Troubleshooting Connection Errors

If an error occurs when you select a connection profile and attempt to establish a connection to the MySQL server named in the profile, a window containing error information will be displayed (see figure).

Figure B.1. Error message dialog



Normally, this means either that the credentials specified in the profile are wrong (wrong username, wrong password), or that you do not have privileges to connect to the MySQL server from the client machine. You can find more information about the error in the Causes of Access denied Errors [http://dev.mysql.com/doc/mysql/en/Access_denied.html] section of the reference manual.

However, another cause of connection failure is that there might be a network problem that prevents you from reaching the machine where the MySQL server is running. For this reason, the error dialog provides a PING button. Clicking this button will send an ICMP PING request to the MySQL server host machine. If the host machine is available over the network, you will see something like this:

```
Reply from 127.0.0.1: Time=0ms TTL=128
Reply from 127.0.0.1: Time=1ms TTL=128
```

This would indicate that the machine is reachable, and that the cause of connection failure is not a network problem.

B.2. Troubleshooting Display Problems

If, under Windows, the controls in the windows look somehow displaced, select Start, Control Panel, Display. Select the Settings tab and click the Advanced... button. Change your DPI settings to Normal (96 DPI).

If, under Linux, messages and captions appear very big, your display resolution is probably set too high. Use `gnome-font-properties` to change your default font to a smaller value or go to `Details` and set your resolution to 72 or some other more appropriate value.

Appendix C. How Connections Are Stored

On Windows, connection profiles are stored in `C:\Documents and Settings\UserName\Application Data\MySQL` directory, where *UserName* is the name of the current Windows user. On Linux, the files are stored in the `~/.mysqlgui` directory. The settings are stored in a file called `mysqlx_user_connections.xml`. The contents of that file look like this:

```
<?xml version="1.0"?>
<user_connections>
  <last_connection>1</last_connection>
  <password_storage_type>2</password_storage_type>
  <user_connection>
    <connection_name></connection_name>
    <username>root</username>
    <hostname>localhost</hostname>
    <port>3306</port>
    <schema></schema>
    <advanced_options/>
    <storage_path></storage_path>
    <notes></notes>
    <connection_type>0</connection_type>
    <storage_type>2</storage_type>
    <password></password>
  </user_connection>
  ...
</user_connections>
```

You can edit the file manually, but take care not to invalidate the XML. When applying changes by editing and saving the file, those changes will show up the next time you open the Connections section of the Options Dialog. You do not need to restart your application for the changes to take effect.

As a database administrator, you may also edit the file according to your preferences, and then copy it to any other machine running the MySQL GUI tools. This makes it easy to have identical connection profiles on all machines, without having to set up those profiles individually. All XML configuration files for the MySQL GUI Tools are cross-platform compatible and can be transferred between Windows and Linux machines.

Appendix D. XML Files Common to the MySQL GUI Applications

On Windows, XML files common to all MySQL GUI applications are stored in the `C:\Documents and Settings\UserName\Application Data\MySQL` directory, where *UserName* is the name of the current Windows user. On Linux, the files are stored in the `~/ .mysqlgui` directory.

- `mysqlx_common_options.xml`: Stores options selected in the Options dialog.
- `mysqlx_user_connections.xml`: This file is described in How Connections are Stored.

Appendix E. XML Files Used by MySQL Query Browser

Besides the common files, MySQL Query Browser uses a number of XML files for internal purposes.

- `mysqlqb_functions.xml`: Stores list of functions and operators for use in the Function Browser.
- `mysqlx_dbm_data_types.xml`: Stores a list of the available MySQL data types for use with auto-completion and with the MySQL Table Editor
- `mysqlx_dbm_charsets.xml`: Stores the available MySQL character sets.
- `mysqlqb_history.xml` : Stores the queries that make up the History Browser
- `mysqlqb_bookmark.xml`: Stores the queries that form the Bookmark Browser

Appendix F. Notes for Translators

You may find that the MySQL GUI tools and/or their documentation are not available in your preferred language. If you would like to translate the software, or its documentation, or participate in that translation, *please contact the MySQL documentation team before starting your translation!*

The MySQL documentation team's address is <docs@mysql.com>. In your mail, please state what you would like to translate (the tools, their documentation, or even the whole MySQL reference manual if you dare), and give some information regarding your background:

- What is your name?
- In which country and city are you located?
- How long have you used MySQL?
- Have you done other translations?
- Will you work alone, or with a group of translators?
- What's your timetable regarding the translation?
- What is your motivation for translating?

We will get back to you as soon as possible.

The format used for both software and documentation is *DocBook XML*. From that base format, all other available formats (HTML, CHM, PDF, etc.) are being generated. It would be a pity if you started translating, say, the HTML version of this documentation, because we will not be able to use it in that format. Also, you might find that someone else has already done (or is in the process of doing) that translation. So, once again, please contact the MySQL documentation team first!

Appendix G. Third-party Software Used by the MySQL GUI Tools

The MySQL GUI tools incorporate PCRE and PNG support through the use of third-party libraries.

G.1. PCRE Library

Regular expression support is provided by the PCRE library package, which is open source software, written by Philip Hazel, and copyright by the University of Cambridge, England. The source for the PCRE library can be found at: <ftp://ftp.csx.cam.ac.uk/pub/software/programming/pcre/> [ftp://ftp.csx.cam.ac.uk/pub/software/programming/pcre/].

G.2. PNG Support

PNG support for the Windows version is provided by the TPNGImage component which is open source software, written and copyright by Gustavo Daud. The source for the TPNGImage can be found at: <http://pngdelphi.sourceforge.net> [http://pngdelphi.sourceforge.net].